INTERNET OF THINGS SUMMARY

(written by Davide Giannubilo – a.y. 22/23)

8. ZigBee

We are focusing on "*Capillary Multi-hop network*" that works with devices within 100 meters.

These networks are highly fragmented by a lot of technologies, protocols and standard:

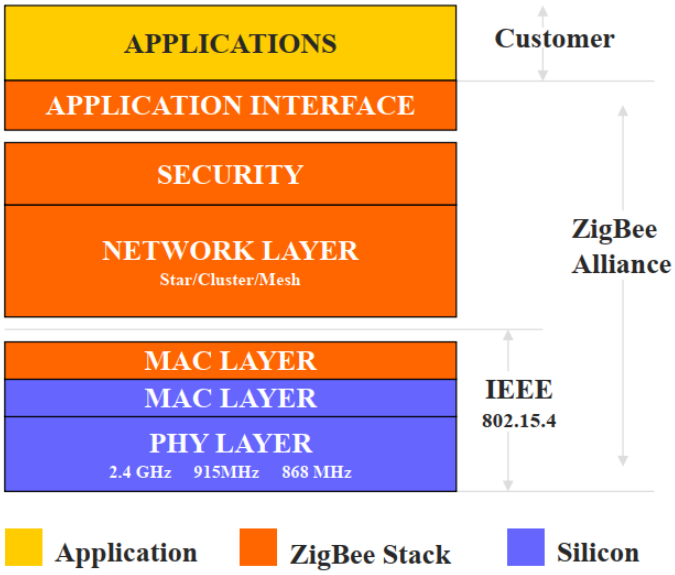
- Proprietary
  - WirelessHART
- Open
  - WiFi, ZigBee, 6LowPAN, THREAD

There are some technologies that are "*application specific*" like, for example, ZWAVE for home automation, and some that are well fitted for everything.

ZigBee is not IP-compliant.

8.1 Main features

- Low-cost hardware and software
- Limited TX range (around 10m)
- Low latency
- High energy efficiency



8.2 Types of device and topologies

- Devices:
  - **Full Function Device (FFD)**:
    - Can send beacons
    - Can communicate with other FFDs

        - Can route frames
        - Can act as PAN coordinator
        - Typically features power supply
    - **Reduced Function Device (RFD)**:
        - Cannot route frames
        - Cannot communicate with other RFDs
        - Can communicate with FFD
        - Runs typically on batteries
    - **PAN Coordinator**
        - Is responsible of a Personal Area Network (PAN)
        - Manages PAN association/de-association
- Topologies
    - Stars with a PAN coordinator in the middle
    - Mesh
    - Cluster-Tree (not in 802.15.4 standard)

8.3 802.15.4 Physical Layer

It offers a lot of features:

- Activation and deactivation of the radio transceiver
- Energy detection (ED) within the current channel
    - Detect energy level for each channel (used to implement scanning functionalities)
- Link quality indicator (LQI) for received packets
- Clear channel assessment (CCA)
    - Used to implement the carrier sense multiple access with collision avoidance (CSMA-CA)
- Channel frequency selection
- Data transmission and reception

Packets are characterized by:

- Preamble, in order to achieve synchronization
- SFD, it is a frame delimiter
- Frame length, in octets (for MAC data frames in the range of 9-127)

<table>
<tr><td colspan="2"></td><td colspan="3">Octets</td></tr>
<tr><td colspan="2"></td><td colspan="2">1</td><td>variable</td></tr>
<tr><td>Preamble</td><td>SFD</td><td>Frame length
(7 bits)</td><td>Reserved
(1 bit)</td><td>PSDU</td></tr>
<tr><td colspan="2">SHR</td><td colspan="2">PHR</td><td>PHY payload</td></tr>
</table>

8.4 802.15.4 MAC Layer

The features of the MAC sublayer are:

- beacon management
- channel access management
- GTS management
- Frame validation
- acknowledged frame delivery
- association, and disassociation
- hooks for implementing application-appropriate security mechanisms.

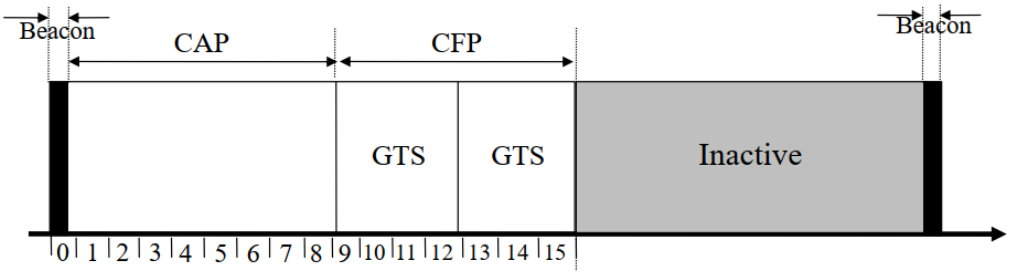
There are 2 operation modes:

1. **Beacon enabled**
    a. PAN coordinator periodically transmits beacons
    b. Usually adopted in star topologies
    c. Slotted CSMA/CA + scheduled transmissions
2. **Non-Beacon enabled**
    a. Uncoordinated access through unslotted CSMA/CA

Beacon enabled

In the beginning, we receive a Beacon message that mark the starting point of a Beacon interval. This interval is divided into two parts:

- active part
    - CAP (Collision Access Part), in which the access is random through CSMA/CA.
    - CFP (Collision Free Part), in which the access is scheduled and divided into two GTS (Guaranteed Time Slot) slots, one for uplink and one for downlink.
- inactive part
    - where the nodes are sleeping.



CSMA/CA

Each device shall maintain three variables for each transmission attempt: **NB**, **CW** and **BE**.

- **NB** is the number of times the CSMA-CA algorithm was required to backoff (initialized to zero before each new transmission attempt)
- **CW** is the contention window length, defining the number of backoff periods that need to be clear of channel activity before the transmission can commence (initialized to 2, only for slotted CSMA-CA).

- **BE** is the backoff exponent, which is related to how many backoff periods a device shall wait before attempting to assess a channel.

Backoff period: duration of 20 symbols.

In general, it works in this way:

1. NB and CW are set equal to 0
2. A transmitting node delays for a random number of backoff periods in $[0, 2^{BE} - 1]$
3. If clear channel assessments (CCA) is idle for CW consecutive backoff periods, the node starts the transmission and waits for an ACK.
4. If the channel is busy, the exponent BE and the number of backoff attempts, NB, are incremented and the procedure is repeated
5. After "too many" (NB_{MAX}) failed retries, the packet is discarded

Transmission procedure (including ACK) must end within a CAP. In case of collision (ACK does not come back), the procedure restarts.

Superframe specification

Bits: 0–3	4-7	8-11	12	13	14	15
Beacon Order	Superframe Order	Final CAP Slot	Battery Life Extension (BLE)	Reserved	PAN Coordinator	Association Permit

- **Beacon Order (BO)**: defines the Beacon Interval (BI)
  - $BI = aBaseSuperframeDuration * 2^{NO}\ symbols$
- **Superframe Order (SO)**: defines the Superframe Duration (SD)
  - $SD = aBaseSuperframeDuration * 2^{SO}\ symbols$
- $aBaseSuperframeDuration = 16\ slots\ x\ 60\ symbols$

Network formation: Scanning

There are two ways:

- Active Scanning (only for FFDs)
  - a beacon request message is sent out to trigger beacon transmission
  - Upon termination of the scanning procedure a PAN ID is chosen
- Passive Scanning (for FFDs and RMDs): similar to Active Scanning but without explicit Beacon Request messages

There are also some extensions of 802.15.4 and they are:

- IEEE 802.15.4e – Slotted channel access
- IEEE 802.15.4g – Amendment for smart utility applications
- IEEE 802.15.4k – Amendment for critical infrastructure monitoring

8.5 ZigBee Network layer

- **Configuring a new device**: this is the ability to sufficiently configure the stack for operation as required.
- **Starting a network**: this is the ability to establish a new network.
- **Joining, re-joining, and leaving a network**: this is the ability to join, re-join or leave a network as well as the ability of a ZigBee coordinator or ZigBee router to request that a device leave the network.
- **Addressing**: this is the ability of ZigBee coordinators and routers to assign addresses to devices joining the network.
- **Neighbour discovery**: this is the ability to discover, record, and report information pertaining to the one-hop neighbours of a device.
- **Route discovery**: this is the ability to discover and record paths through the network, whereby messages may be efficiently routed.
- **Reception control**: this is the ability for a device to control when the receiver is activated and for how long, enabling MAC sub-layer synchronization or direct reception.
- **Routing**: this is the ability to use different routing mechanisms such as unicast, broadcast, multicast, or many to one to efficiently exchange data in the network.

We have three types of devices:

- ZB Coordinator (FFD)
- ZB Router (FFD)
- ZB End-Device (RFD o FFD)

And two types of routing techniques:

- Ad-hoc On-demand Distance Vector (AODV)
- Cluster Tree Algorithm

Cluster Tree Algorithm: Tree formation

- A FFD scans the available channels through the proper functionalities at the lower layers
    - Chooses a channel (e.g., the least interfered)
    - Sets the PAN identifier
    - Sets its own Network Address to 0 (Coordinator)
- Other devices may now associate to the coordinator through the lower layer association procedures

Associated devices may be:

- ZB Router (only FFD): may let other devices to associate to the network
- ZB End-Device

Address Assignment (16 bits short addresses) is performed jointly with association. Each parent device (PAN coordinator, ZB router) assigns groups of addresses to its children (other ZB routers, ZB end devices).

The ZigBee coordinator fixes:

- the maximum number of routers (R_m)
- end-devices (D_m) that each router may have as children
- the maximum depth of the tree (L_m).

Address assignment Rule

The size $A(d)$ of the range of addresses assigned to a router node at depth $d < L_m$ is defined by:

$$A(d) = \begin{cases} 1 + D_m + R_m & if\ d = L_m - 1 \\ 1 + D_m + R_m A(d+1) & if\ 0 \leq d < L_m - 1 \end{cases}$$

A mote at level d is assigned addresses in range $[x,\ x + A(d) - 1]$. It will assign:

- $[x + (i-1)A(d+1) + 1,\ x + iA(d+1)]$ to its *i-th* router child ($1 \leq i \leq R_m$)
- $x + R_m A(d+1) + j$ to its *j-th* end-device child ($1 \leq j \leq D_m$).

Tree-Based Routing

- If destination address is one of children end devices: route directly
- Else if destination address belongs to one of children routers' addresses set: send to corresponding children router
- Else send to parent node

There could be some problems like, the tree is optimized, the tree is unbalanced, etc.

AODV Routing

- A node willing to send to a destination broadcast a Route Requests (RREQ) message aka shout "where's the destination"
- RREQ messages are flooded by receiving nodes
  - relay shouting
- When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source
  - stores "who shouted at me"
- When the intended destination receives a Route Request, it replies by sending a Route Reply
  - shouts back "It's me"
- Route Reply travels along the reverse path set-up when Route Request is forwarded
  - shouting travels back the same route

The **routing table** is made in this way:

- Destination Address: 16-bit network address of the destination
- Next-hop Address: 16-bit network address of next hop towards destination
- Entry Status: One of Active, Discovery or Inactive

Routing Discovery Table:

- RREQID Unique ID (sequence number) given to every RREQ message being broadcasted

- Source Address: Network address of the initiator of the route request
- Sender Address: Network address of the device that sent the most recent lowest cost RREQ
- Forward Cost: The accumulated path cost from the RREQ originator to the current device
- Residual Cost: The accumulated path cost from the current device to the RREQ destination

Entries of RT and RDT have validity time-outs.

Routing cost

The cost for path P composed of $L-1$ links is defined as:

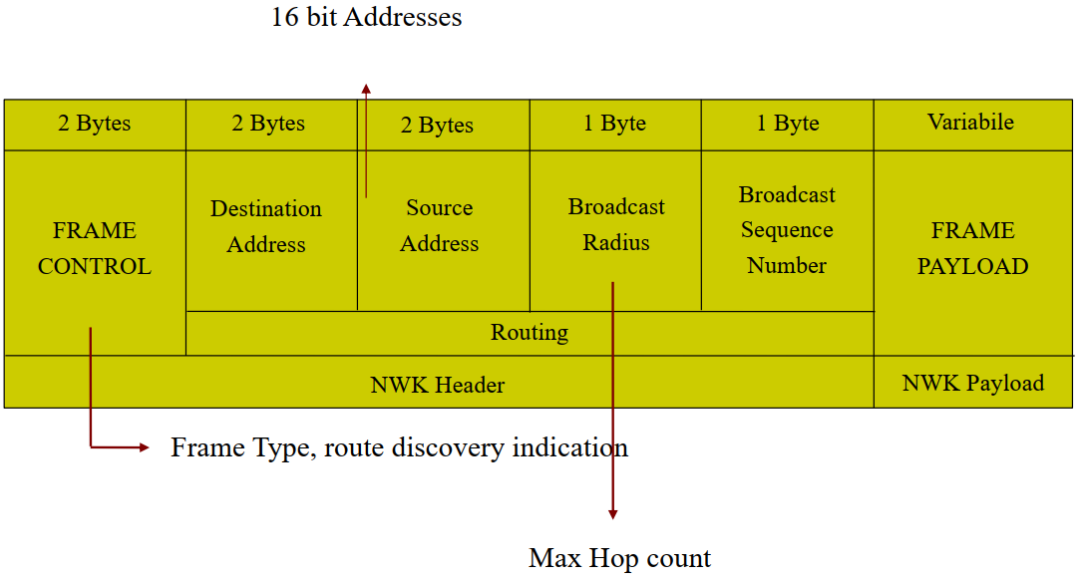
$$C\{P\} = \sum_{i=1}^{L-1} C\{[D_i, D_{i+1}]\}$$

ZigBee standards "*suggests*" the following form for the cost of the generic link l:

$$C\{l\} = \begin{cases} 7, \\ \min\left(7, round\left(\frac{1}{p_l^4}\right)\right) \end{cases}$$

P_l is the packet reception rate over link l.

Frame format



8.6 ZigBee application profiles

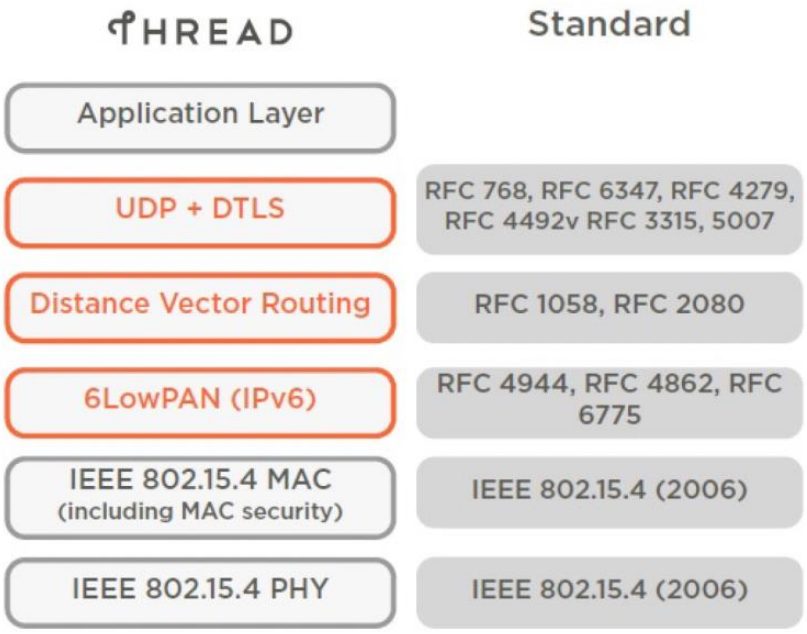
We need:

- A common language for exchanging data
- A well-defined set of processing actions
- Device interoperability across different manufacturers
- Simplicity and reliability for the end users

9. THREAD & 6LowPAN

9.1 THREAD protocol

Thread is an IPv6-based, low-power mesh networking technology for Internet of things (IoT) products. It is built using open and proven standards.



We have two types of devices:

- **Full Thread Device**
  - Router, route packets, always active
  - Leader, elected role of one router, always active
  - Routing-eligible End Device (REED), can become router
  - Full End Device (FED)
- **Minimal Thread Device**
  - Minimal End Device (MED), always on
  - Sleepy End Device, duty cycling
  - Synchronized Sleepy End Device, duty cycling at scheduled time

We also have boarder router.

The topology can change over time because, for example, REED devices can be upgraded to operate as a router, or, when a Router has no children, it can downgrade itself and operate as an End Device.

Thread Leader is dynamically self-elected for fault tolerance, aggregates and distributes network-wide configuration information.

A **Border Router** is a device that can forward information between a Thread network and a non-Thread network (for example, Wi-Fi).

How is a THREAD network identified?
It has:

- 2-byte Personal Area Network ID (PAN ID)
- 8-byte Extended Personal Area Network ID (XPAN ID)
- A human-readable Network Name

THREAD Network Discovery

1. The device broadcasts an 802.15.4 Beacon Request on a specific Channel.
2. Any Routers (or REED) in range broadcast a Beacon that contains their Thread network PAN ID, XPAN ID, and Network Name.
3. The device repeats the previous two steps foreach Channel.

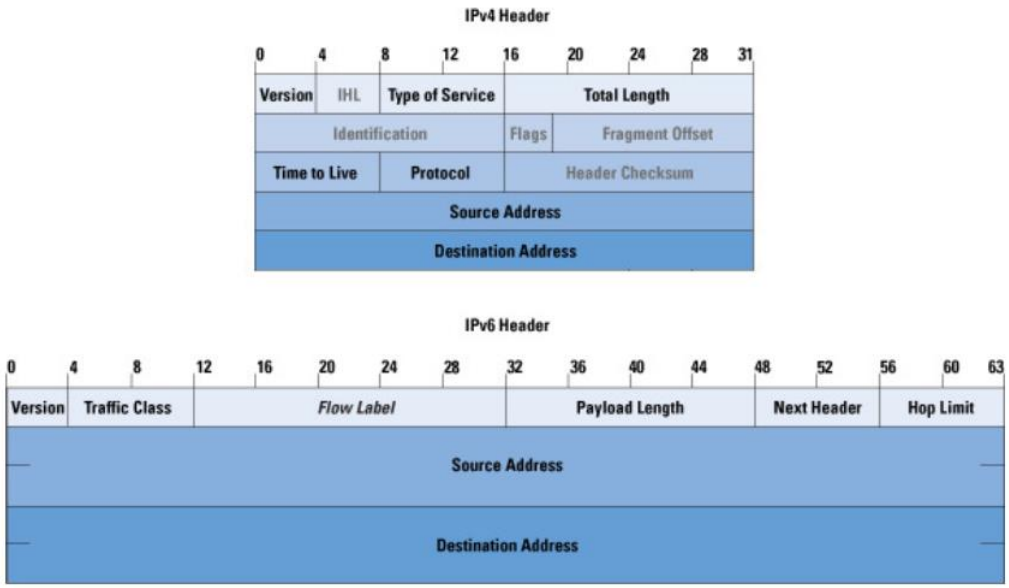
THREAD: Joining a network

1. The child sends a multicast Parent Request to all neighbouring Routers and REEDs in the target network.
2. All neighbouring Routers and REEDs (if the Parent Request Scan Mask includes REEDs) send Parent Responses with information about themselves.
3. The Child chooses a Parent device and sends a Child ID Request to it.
4. The Parent sends a Child ID Response to confirm link establishment. (source/destination RLOC, info about THREAD network)

IPv6

IPv6 (RFC 2460) = the next generation Internet Protocol. It uses 128-bit addresses, theoretically allowing 2^{128}, or approximately $3.4\ x\ 10^{38}$ total addresses.

It provides a stateless auto-configuration and simple routing and address management.



128-bit IPv6 address = 64-bit prefix + 64-bit Interface ID (IID).

- The 64-bit prefix is hierarchical, it identifies the network you are on and where it is globally

- The 64-bit IID identifies the network interface, it must be unique for that network and typically is formed stateless from the interface MAC address (called Stateless Address Autoconfiguration (RFC2462)).

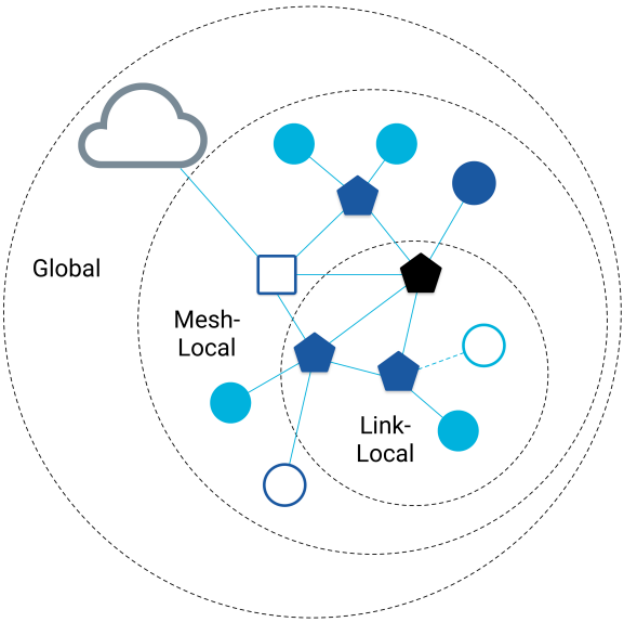
There are different kinds of IPv6 addresses depending on scope and context.

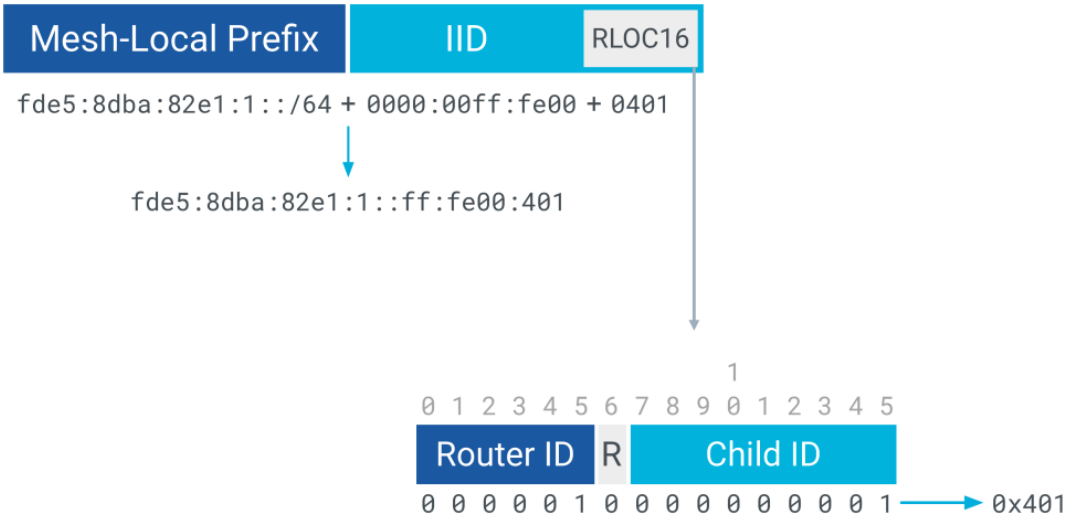
- Loopback (0::1) and Unspecified (0::0)
- Unicast with global (e.g. 2001::) or link local (FE80::) scope
- Multicast addresses (starts with FF::)
- Anycast addresses (special-purpose unicast address)

IPv6 in THREAD

We have three scopes:

- Link local: interfaces reachable by direct transmission fe80::/16
- Mesh local: interfaces reachable within the same Thread network fd00::/8
- Global





Other THREAD addresses:

- Link Local Address (LLA)
    - Used to discover neighbours, configure links, and exchange routing information
    - Not a routable address
    - Based on IEEE802.15.4
- Mesh Local EID (ML-EID)
    - Random, chosen after commissioning is complete
    - Does not change as the topology changes
    - Should be used by applications
- Anycast Locator (ALOC)
- Global Unicast Address (GUA)
    - A public IPv6 address
    - IID manually assigned or DHCP
    - Always has a prefix of 2000::/3

9.2 6LowPAN

It was created with the intention of applying the Internet Protocol (IP) even to the smallest devices, enabling low-power devices with limited processing capabilities to participate in the Internet of Things.

It is characterized by:

- Efficient header compression.
- Fragmentation of the IPv6 packet from 1280 byte to 127 byte as requested by 802.15.4 based networks.
- Auto-configurating networks through network discovery.
- Unicast, multicast and broadcast messages.

Header compression

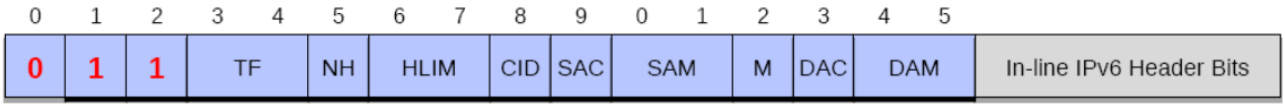
6LoWPAN is characterized by the compression of headers for the use of protocols characteristic of the internet, such as IPv6 and UDP, on PAN. Key features of this process include:

- compression is always stateless and flow-independent, therefore with the possibility to decompress a package independently from the others belonging to the same flow.
- The version is always set to 6.
- Traffic class and flow label are always set to 0, as the packages belong to the same type of traffic.
- Payload length and derived from L2 headers.
- Source and destination addresses can be elided or compressed into based on the broadcast context.

We talk about *Header Compression* header to define the set of fields necessary to define how the compression will happen in the relative fields of the package:

- TF (Traffic Class and Flow Label)
    - 0: Carried Inline (ECN+DSCP+Flow), 1: ECN+Flow, 2: ECN+DSCP, 3: All zero

- NH (Next Header compression)
  - 0: Carried Inline, 1: Next Header is compressed
- HLIM (Hop Limit = Inline, 1, 64, 255)
  - 0: Carried Inline, 1: 1, 2: 64, 3: 255
- CID (Context Identifier Extension)
  - 0: No 1-byte CID identifier, 1: 1-byte identifier follows
- SAC/DAC (Source/Destination Address Compression)
  - 0: Stateless, 1: Context-based
- SAM/DAM (Source/Destination Address Mode)
  - 0: 16 bytes inline, 1: 8 bytes inline, 2: 2 bytes inline, 3: elided
- M (Multicast Destination)
  - 0: Destination is not multicast, 1: Destination is multicast



Also the **UDP header** is compressed.

Fragmentation

Since IPv6 requires a minimum of 1280 bytes, we need to divide it in order to work on 802.15.4 networks which have a maximum payload of 127 bytes. For this reason, a fragmentation header is used, and it is composed by:

- *dgram_size*: size of the fragment in bytes.
- *dgram_tag*: fragmentation ID (common to all fragments).
- *dgram_offset*: fragmentation offset (word of 8 bytes). Elided in the 1° fragment.

Although fragmentation is inevitable, it is not used in practice and should be avoided because it would lead to a significant drop in performance. For example, if we lost a fragment, we need to resend everything.

Routing

We have two different classes of algorithms:

- **Distance-vector**, links are associated with cost, used to find the shortest route. Each router along the path store local next-hop information about its route table.
- **Link-state**, each node acquires complete information about the network, typically by flooding. Each node calculated a shortest-path tree calculated to each destination.

We also have two types of signalling:

- Proactive
  - Routing information acquired before it is needed.
- Reactive
  - Routing information discovered dynamically when needed.

An important factor is the route metrics. The most common are referred to:

- Node
  - Residual energy
  - CPU, Storage, WorkLoad, Battery/Mains
- Link
  - Throughput (local/global metric)
  - Latency (local/global metric)
  - Reliability (local/global metric)
    - Expected Transmission Count (ETX)
      "*The average number of packet transmissions to successfully transmit a packet*"
    - Link Quality Level (LQL)
- Hop Count

Imagine having a THREAD routing, proactive and distance vector, each router has a routing database made by:

- *Route id set*: list of routers ID and SN
- *Link set*: list of current/recent neighbours `<l_router_id, l_link_marging, l_incoming_quality, l_outgoing_quality, l_age>`
- *Route set*: routing table `<destination, next_hop, route_cost>`

There are also periodic signalling messages made by:

- Router ID, incoming_quality, outgoing_quality, route cost, sequence_number

10. Bluetooth

- Radio technology
- Low cost
- Small range (10-20 m)
- Low complexity
- Small size
- ISM 2.4 GHz band
- Created by an industrial consortium
- Only the first two levels have been standardized by IEEE 802.15.1

This technology is used for:

- Headset/Audio
- Data transfer
- Connectivity sharing
- Proximity marketing (Bluetooth Low Energy – BLE – only)

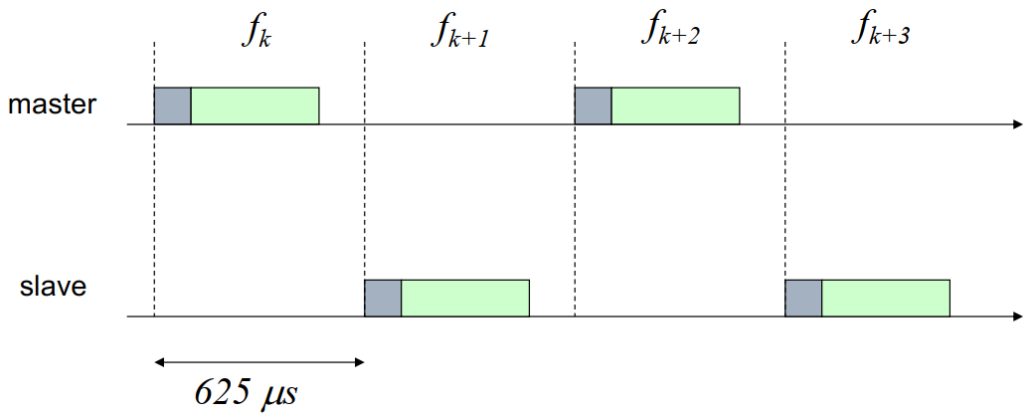
10.1 Physical layer

- ISM band at 2.4 GHz
- 79 (23 in France and Japan) channels spaced of 1 MHz (2402-2480 MHz)
- Modulation G-FSK (1 Mb/s)

There are different device classes:

Class	Power (mW)	Power (dBm)	Range (approx.)
Class 1	100	20	100 m
Class 2	2,5	4	10 m
Class 3	1	0	1 m

- Frequency Hopping (*FH*)
- 1600 hops/s (625 µs per hop)
- The *FH* sequence is pseudo random and determined by the clock and the address of the "*master*" device that regulates the access to the channel.
- The other devices are "*slaves*" and follows the sequence f_k defined by the master.

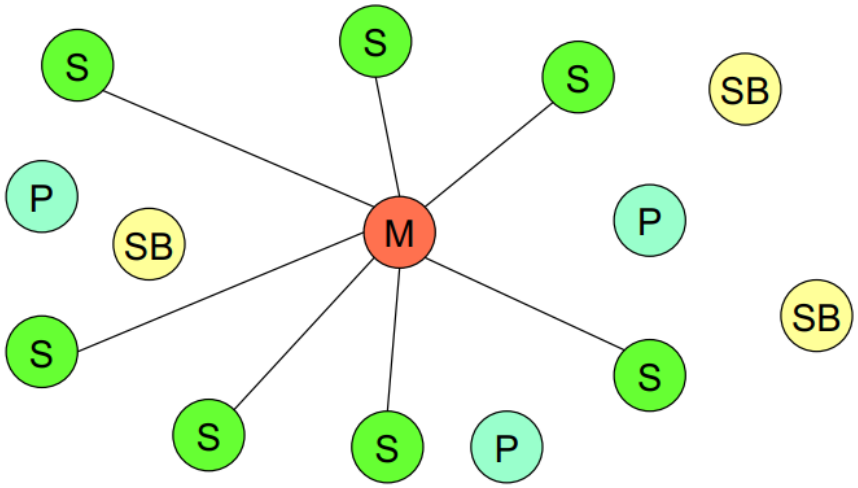


- The numbering of the slots is defined by the clock of the master
- The sequence is given by the master ID and a generation algorithm

It is possible to transmit packet with duration of 1, 3 or 5 intervals. So, imagine having a master that uses 3-slot packet f_k, the next packet sent from the slave will be $f_k + 3$ and so on.

10.2 Piconet

The simplest network architecture defined in Bluetooth is called "***piconet***". It is an ad hoc network composed of 2 or more devices.
A device acts as *master* and the other as *slaves*. Communication can take place only between master and slave and not directly between slaves and up to 7 slaves can be active in a piconet. The others can be in *Stand-by* (not part of the piconet) or *Parked* (part of the piconet but not active, up to a maximum of 256 devices).



Addresses:

- MAC address of 48 bits
- AMA (Active Member Address) 3 bits
- PMA (Parked Member Address) 8 bits

Types of connections:

1. **SCO (Synchronous Connection Oriented)**
    a. Fixed rate bi-directional connection (circuit)
    b. FEC for improving quality
    c. Rate of 64 Kbit/s
2. **ACL (Asynchronous ConnectionLess)**
    a. Packet switched connection shared between master and active slaves based on a polling access scheme
    b. Several options for packet formats and physical layer codes (1, 3, 5 slots)
    c. Rate up to 433.9 Kbit/s symmetric (using 5-slot packets in both directions) and 723.2/57.6 Kbit/s asymmetric (using 5-slot packets in one direction and 1-slot packets in the other)

11. RFID

RFID, which stands for Radio Frequency Identification, was born as a technology for tracking and identifying objects wirelessly. It is used in a variety of applications, including access control, asset tracking, and even in contactless payment systems.

In order to use this technology we need:

- a reader
- a tag

11.1 Tag

There are different types of tags differentiated by the **different power supply**:

- passive
    - operational power scavenged from reader radiated power
    - short range | low cost | self-sustaining
    - 1-bit tags, when tag gets close to reader, current variation is perceived at the reader
- semi-passive
    - operational power provided by a battery
    - medium range (tens of meters) | need a battery | average cost | long life
- active
    - operational power provided by a battery and there is a transmitter built into the tag
    - high range (hundreds of meters) | need a battery | limited lifetime

Tags with storage memory space can be read only and read/write.

Class 0	UHFI read-only, preprogrammed passive tag
Class 1	UHF or HF; write once, read many (WORM)
Class 2	Passive read-write tags that can be written to at any point in the supply chain
Class 3	Read-write with onboard sensors capable of recording parameters like temperature, pressure, and motion; can be semipassive or active
Class 4	Read-write active tags with integrated transmitters; can communicate with other tags and readers
Class 5	Similar to Class 4 tags but with additional functionality; can provide power to other tags and communicate with devices other than readers

Of course the tags can be attached to anything.

11.2 Reader

An RFID reader is an electronic device that is used to communicate with RFID tags and retrieve the information stored on them. The reader consists of a radio frequency module, a microcontroller, and an antenna.

RFID readers come in a variety of sizes and shapes, depending on the specific application. Some readers are handheld and portable, while others are mounted on a wall or a fixed position. The

range of the reader's antenna also varies, with some readers capable of communicating with tags over several meters, while others have a much more limited range.

11.3 Performance measures

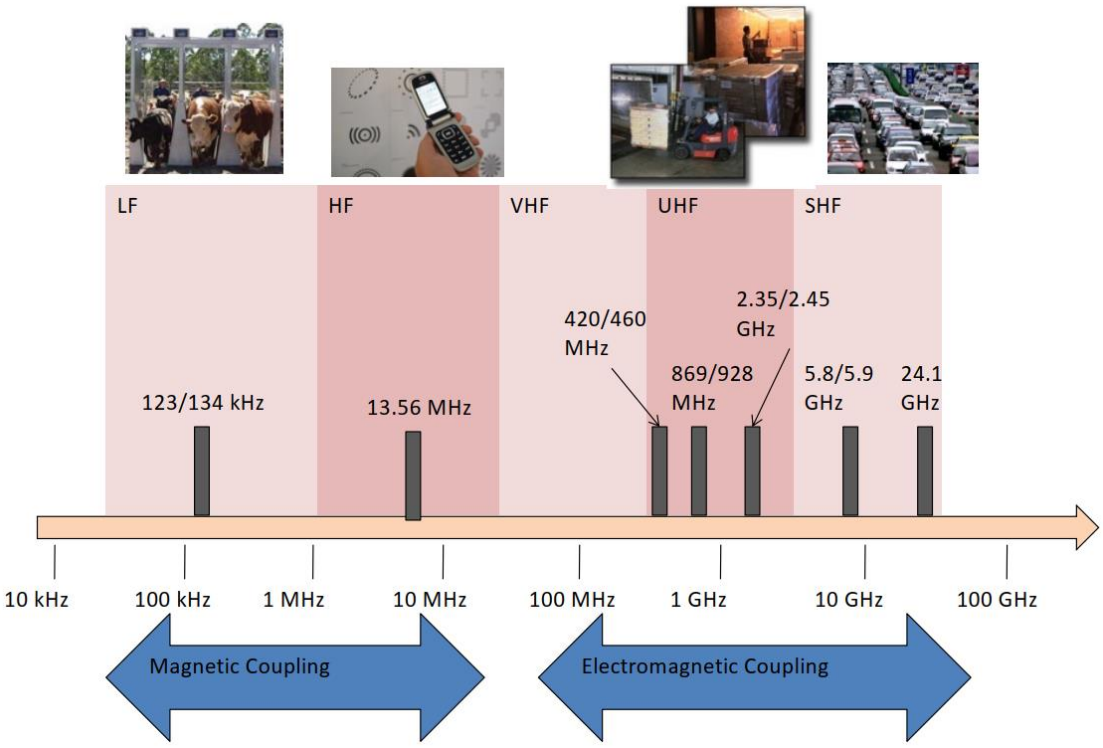
We have three indicators:

- Reading range
- Throughput
- Robustness

Performance depends on:

- Carrier frequency
- Emitted power
- Environment (propagation conditions)
- Concurrency (# of tags to be read simultaneously)

11.4 RFID standards

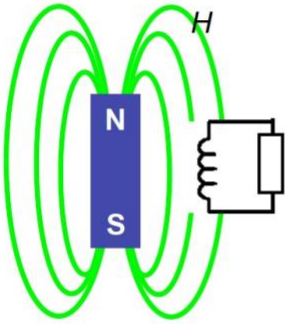


RFID has different standards available based on the fields of application:

- animal identification
- cards and personal identification
- containers ID

Physical communication

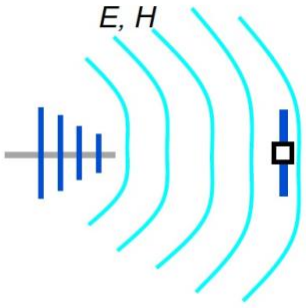
Near Field Model (HF)



125 kHz – 13.56 MHz

125 kHz	0.05 m
13.56 MHz	0.5 m

Far Field Model (UHF)



400 MHz – 2450 MHz

860 - 930 MHz	4-10 m
2450 MHz	1 m

RFID HF		RFID UHF	
PROs	**CONs**	**PROs**	**CONs**
Almost immune to the environment	Limited reading range	High reading range	High impact of the environment
Moderate cost	Very sensitive to orientation	High bit rate	
In LF/HF magnetic field is scarcely affected by dielectric materials			

11.5 RFID collision

Imagine having a reader that interrogate some tags, they will answer to it, but we need a mechanism that will handle them correctly without collisions.

There are some of these mechanisms:

- Vertical classification
  - Slotted ALOHA
  - Dynamic Frame ALOHA
  - Tree-based → Binary tree
- Horizontal classification
  - Centralized/Distributed
  - Type of Channel Feedback (S,C,0)

Tag arbitration efficiency

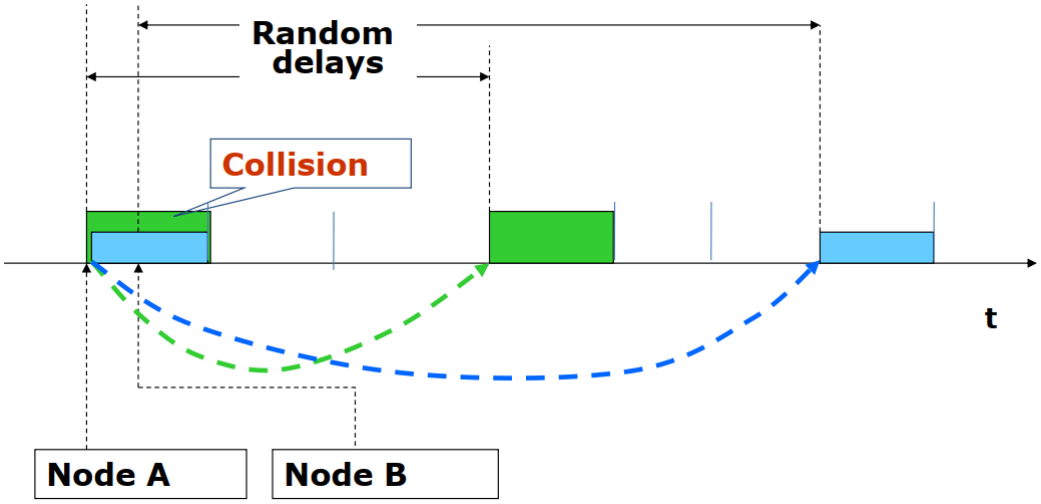
The efficiency is commonly defined as the tag population size, N, over the length of the arbitration period $L(N)$.

$$Efficiency\ \eta = \frac{N}{L_N}$$

Slotted ALOHA protocol

In this protocol we do not have channel feedback, only the ACK, and the time is slotted. It works in this way:

- The first packet in the transmission queue is transmitted in the first available slot.
- If the ACK does not come, the transmission is re-attempted after a random number of slots X



Something about **performance**.

We have traffic G distributed according to Poisson process, packets arrival is a Poisson point process with parameter λ and transmission lasts T.
So, $G = T * \lambda$

Now, the probability P_s for a packet transmission to be successful is the probability that no other packet arrives in the previous slot.

$$P_s = P[N(t-T, t) = 0] = e^{-G}$$

So,

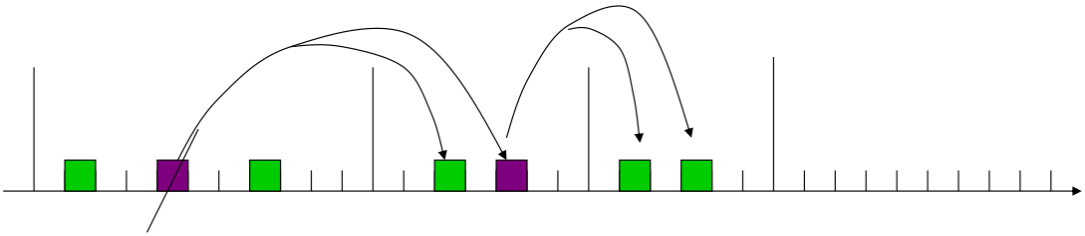
$$Throughput\ S = G * e^{-G}$$

Single Frame ALOHA

This is an extension of the ALOHA protocol where nodes are allowed to transmit once every frame.
How it works?

- A frame composed of r slots.
- Every tag chooses a slot in the frame.

- If transmission is failed, retry at next frame.



$$The\ average\ throughput\ is\ E[S] = n\left(1 - \frac{1}{r}\right)^{n-1}$$

$$The\ efficiency\ is\ \eta = \frac{E[S]}{r} = n\frac{1}{r}\left(1 - \frac{1}{r}\right)^{n-1}\ which\ is\ maximum\ for\ r = n$$

Multiple Frames ALOHA

The FA efficiency depends on the initial tag population N, the current backlog n and the frame size r.

Current Frame size r is dynamically set to the current backlog n → Dynamic Frame Aloha

$$Efficiency\ \eta = \frac{N}{L_N}$$

The average tag resolution process can be recursively calculated as:

$$L_N = r + \sum_{i=0}^{n-1} P(S = i)L_{n-i}$$

which leads to:

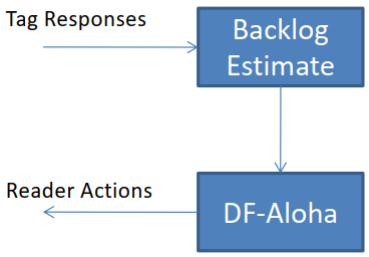
$$L_n = \frac{r + \sum_{i=0}^{n-1} P(S = i)L_{n-i}}{1 - P(S = 0)}$$

Schoute's estimation

We have some problems because initial population N and backlogs n are not known in advance.

Tag arbitration is composed of two modules:

- Backlog Estimation Module: to provide an estimate of the backlog n_{est}
- Collision Resolution: run Frame Aloha with $r = n_{est}$



2 assumptions needed:

- any procedure is able to keep the frame size r equal to the current backlog n
- the number of terminals transmitting in a slot is approximated by a Poisson process with intensity $\lambda = 1[terminal/slot]$

The average number of terminals in a collided slot can be consequently calculated as:

$$H = \frac{(1 - e^{-1})}{(1 - 2e^{-1}} = 2.39$$

The backlog is estimated as:

$$n_{est} = round(Hc), being\ c\ the\ number\ of\ collided\ slots$$